

Совместимость устройств Интернета вещей на железнодорожном транспорте

О. Г. Евдокимова¹, С. М. Куценко², А. С. Галкин³

Петербургский государственный университет путей сообщения Императора Александра I

¹evdokimova@pgups.ru, ²kutsenko@pgups.ru, ³galkinsasha311200@mail.ru

Аннотация. Принимая во внимание сегодняшнее разнообразие устройств, подключенных к сети интернет, большое количество производителей, возникают проблемы взаимодействия между объектами Интернета вещей. Целью данной статьи является анализ беспрепятственного взаимодействия IoT (Internet of Things, Iot) устройств с облачными платформами, независимо от различных производителей, технологий и систем.

Ключевые слова: Интернет вещей, информационная модель, совместимость устройств

I. ВВЕДЕНИЕ

Главным фактором развития IoT в настоящее время является быстрое увеличение числа терминальных устройств, работающих на базе этой технологии (14,4 млрд устройств в 2022 году [1]). Это также является его главной проблемой, так как рост количества используемых «умных вещей», несмотря на многочисленные преимущества их применения, обнаруживает также недостатки. Так, возникают трудности в области управления потоками данных большого объема, а также принятия оптимальных решений при развёртывании систем IoT и обеспечении взаимодействия устройств внутри сети. Последнее ещё более осложняется конкуренцией между производителями, занятыми в данной сфере [5]. Дополнительно ситуацию также усугубляет быстрое развитие и видоизменение технологий беспроводной связи. Как результат, на сегодняшний день рынок IoT очень неоднороден, наполнен большим количеством устройств, не всегда способных взаимодействовать между собой. Это обстоятельство замедляет развитие отрасли в целом. Не способствует ему и отсутствие единых стандартов, вкуче с ограниченным практическим опытом совместного использования разных устройств являющееся на данный момент одной из основных проблем.

II. СОЗДАНИЕ СТАНДАРТОВ

Одним из вариантов решения является создание консорциумов и союзов компаний для формирования единых стандартов взаимодействия, безопасности и связи устройств Интернета вещей [2, 5]. Наиболее влиятельными консорциумами являются:

- AllSeen Alliance. В состав союза входят более 50 компаний таких как LG, Microsoft, HTC, Haier, TP-Link.

The Open Interconnect Consortium (OIC) – консорциум, сформированный корпорацией Intel. В него

вошли такие компании как Atmel, Dell, Broadcom и Samsung.

- Lora Allience. Тесное сотрудничество более чем 500 участников помогает продвигать развитие протокола LoRaWAN.

Анализ текущего состояния процесса стандартизации в сфере Интернета вещей показал, что работа по решению проблем активно ведётся на уровне крупных корпораций, которые являются ведущими производителями IoT устройств [2, 5]. Но говорить о полном устранении проблемы отсутствия беспрепятственного взаимодействия разнородных устройств пока что нельзя.

III. СОВМЕСТИМОСТЬ ВСЕХ УСТРОЙСТВ

Иным способом обеспечения совместимости устройств Интернета вещей является технический подход к стандартизации на уровне семантической абстракции с помощью общих, не зависящих от технологии информационных моделей. В качестве связующего звена между устройствами в статье представлен проект Eclipse Vorto из семейства Eclipse IoT [3].

Eclipse Vorto – это проект с открытым исходным кодом, который можно использовать для создания абстрактных описаний устройств (информационных моделей) и управления ими. В чем особенность: проект не является попыткой установить стандарт, включающий все существующие технологии. Вместо этого Eclipse Vorto сознательно выбирает подход, который не требует стандарта, а скорее предоставляет полезный инструментарий для интеграции интеллектуальных устройств с платформами или приложениями Интернета вещей, снижения соответствующих затрат на разработку и обеспечение совместимости [4]. Vorto занимается вопросами абстрагирования, описания и интеграции устройств. Так называемые информационные модели абстрактно описывают свойства и возможности устройств. Этими информационными моделями можно управлять и обмениваться через репозиторий Vorto. Производители устройств могут публиковать информационные модели, а пользователи могут загружать их, а затем использовать генераторы кода для создания исходного кода для конкретной платформы, что значительно упрощает интеграцию устройства.

Как именно Eclipse Vorto упрощает обработку описаний устройств? Vorto опирается на четыре столпа:

- Набор инструментов Vorto IoT.

- Мета модель.
- Репозиторий (хранилище) Vorto.
- Генератора кода.

A. Набор инструментов Интернета вещей

С набором инструментов IoT Eclipse Vorto предоставляет редактор DSL (предметно-ориентированный язык) на основе Eclipse XText [4]. В дополнение к базовым функциям, таким как подсветка синтаксиса, автозаполнение, поддержка контента и проверка, набор инструментов IoT также предлагает интеграцию с репозиторием Vorto, хранилищем, размещенным в облаке, для управления информационными моделями. Простота языка, который также может быть прочитан и понятен не только разработчиками программного обеспечения, имеет важное значение. Для иллюстрации приведем пример использования инструмента Eclipse Vorto при описании функционального блока, позволяющего осуществлять контроль и мониторинг такого параметра как местоположение («location»):

```
namespace org.eclipse.vorto.example
version 1.0.0
displayname "Location"
description "The location of an object."
category demo
entity Location {
mandatory latitude as float with
{ MeasurementUnit : LocationUnits.Degree, readable :
true }
<MIN -90, MAX 90> "The latitude of the position."
mandatory longitude as float with
{ MeasurementUnit : LocationUnits.Degree, readable :
true }
<MIN -180, MAX 180> "The longitude of the
position."}
```

B. Мета модель

В Eclipse Vorto описания устройств можно создавать на основе так называемых информационных моделей (ИМ). Основой для информационных моделей является метамодель Vorto, определяющая взаимосвязь между типами данных, функциональными блоками (ФБ), информационными моделями и их структуру. Технически метамодель основана на Eclipse Modeling Framework (EMF) [3] и гарантирует, что все модели являются машиночитаемыми и, следовательно, валидируемыми, а также переводимыми. Самой крупной единицей метамодели является информационная модель, которая в конечном итоге представляет описание устройства. Информационная модель (рис. 1) включает в себя различные многократно используемые единицы [4] такие как: функциональные блоки, типы данных (ТД) и перечисления (П). Наименьшими единицами модели являются типы данных и перечисления. Они могут быть использованы для снятия информации, например, о положении устройства в комнате, состоянии открытия/закрытия двери или интенсивности света. Типы данных могут ссылаться друг на друга. Функциональные блоки описывают пакеты функций,

которые могут быть назначены компоненту, например – аккумулятору, датчику освещенности или выключателю. Функциональным блокам присваиваются свойства (атрибуты) и поведение (операции).

Представляется, что независимая от технологии абстракция устройства создает квазистандарт: информационная модель может быть переведена в различные форматы. Конкретные экземпляры модели в конечном итоге служат источником для автоматической генерации кода и, таким образом, в качестве основы для интеграции в различные платформы.

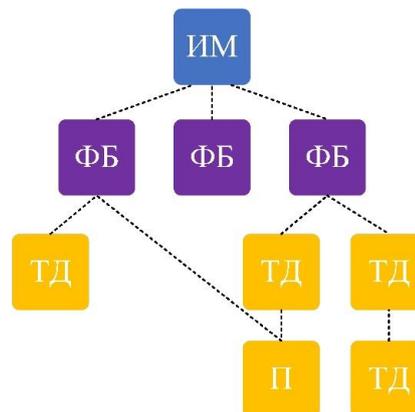


Рис. 1. Мета модель Eclipse Vorto

C. Генератор кода

Генератора кода обеспечивает простой способ перевода информационной модели в другой формат. Генератор предлагает программные интерфейсы, которые позволяют удобно и легко запускать модели и передавать данные с помощью шаблонов. Шаблоны основаны на Eclipse XTend. С их помощью выходные данные могут быть сгенерированы в различных форматах, будь то код или формат документации. С одной стороны, генераторами кода можно управлять непосредственно из набора инструментов IoT, но они также могут работать как микросервис и управляться напрямую через репозиторий через REST. Преимущество: с помощью микросервиса генератора кода можно также обслуживать сценарии приложений, в которых код должен генерироваться во время выполнения.

D. Хранилище Vorto

Описания устройств могут централизованно храниться, управляться и предоставляться через репозиторий Vorto (рис. 2) [3]. Пользовательский веб-интерфейс репозитория предлагает возможность поиска моделей, их загрузки или перевода в другие форматы. Кроме того, пользователи могут писать комментарии непосредственно к модели через репозиторий. Это облегчает создание модели и упрощает обсуждение возможных улучшений. В текущей версии репозиторий Vorto размещен в Eclipse. Он также служит краеугольным камнем для построения экосистемы по теме описаний устройств. Через программный интерфейс REST (VortoRAPI) репозиторий также предоставляет все функции, которые обеспечивают

беспрепятственную интеграцию в существующие инфраструктуры. Генераторы кода также могут быть размещены в этой инфраструктуре (VortoGen).

Vorto Repository Search Generators About Vorto REST API

Filter Information Models Search Clear

Search Results: 51

Type	Name	Namespace	Version	Description
Information Model	Battery	examples.functionblockmodels.metering	1.0.0	Function block model for describing key functionalities of a battery device.
Information Model	TemperatureSensor	examples.functionblockmodels.sensors	1.0.0	Function block model for describing a temperature sensor.
Information Model	MagnetSwitch	examples.functionblockmodels.sensors	1.0.0	Function block model for MagnetSwitch.
Information Model	Accelerometer	examples.functionblockmodels.sensors	1.0.0	Measures the speed and 3 axis position.
Information Model	DoorContact	examples.functionblockmodels.sensors	1.0.0	Function block model for describing a door contact.
Information Model	MotionDetector	examples.functionblockmodels.sensors	1.0.0	Function block model for describing a motion detector.
Information Model	LightSensor	examples.functionblockmodels.sensors	1.0.0	Function block model for LightSensor.
Information Model	Gyroscope	examples.functionblockmodels.sensors	1.0.0	Function block model represents Gyroscope.
Information Model	HumiditySensor	examples.functionblockmodels.sensors	1.0.0	Function block model for describing an humidity sensor.
Information Model	PressureSensor	examples.functionblockmodels.sensors	1.0.0	Function block model for describing a pressure sensor.
Information Model	Lamp	examples.functionblockmodels.illumination	1.0.0	Function block model for describing a simple lamp.
Information Model	Dimmer	examples.functionblockmodels.actuators	1.0.0	Function block model for describing a simple dimmer.
Information Model	BinarySwitch	examples.functionblockmodels.actuators	1.0.0	Function block model for describing a simple switch.
Data Type	SwitchState	examples.datatypes.state	1.0.0	Enum representing the states of a switch.
Data Type	DoorState	examples.datatypes.state	1.0.0	Enum for describing the state of a door.

Рис. 2. Хранилище Eclipse Vorto предоставляет обзор всех доступных моделей

E. Потенциальные пользователи Vorto

С описанными компонентами Eclipse Vorto предлагает набор инструментов для самых разных сценариев в области разработки IoT. Выделим три целевые группы [4], для которых Eclipse Vorto обещает значительную ценность (рис. 3).

F. Производитель устройства

Eclipse Vorto позволяет производителям легко интегрировать свои устройства в существующие инфраструктуры. Например, если конечный пользователь использует систему умного дома определенного производителя, в какой-то момент он может захотеть интегрировать устройства другого производителя в свою существующую инфраструктуру. Традиционно для этой цели производители предоставляют документацию о технических характеристиках своих устройств. Заказчик часто получает документацию в виде PDF или другого распространенного формата документации. С помощью Eclipse Vorto производители могут создавать технологически независимую информационную модель и использовать ее для подробного описания устройства. Информацию из этого описания теперь может быть преобразована как в техническую документацию, так и в исходный код. Это значительно упрощает интеграцию в целевую платформу. Преимущество независимой от технологий информационной модели: производителям устройств больше не нужно беспокоиться о создании реализаций своих устройств для всех возможных целевых платформ. Это экономит время и, следовательно, затраты. Вместо этого производители могут публиковать технологически независимые информационные модели, которые пользователи затем могут легко преобразовывать для соответствующей целевой платформы с помощью генераторов кода.

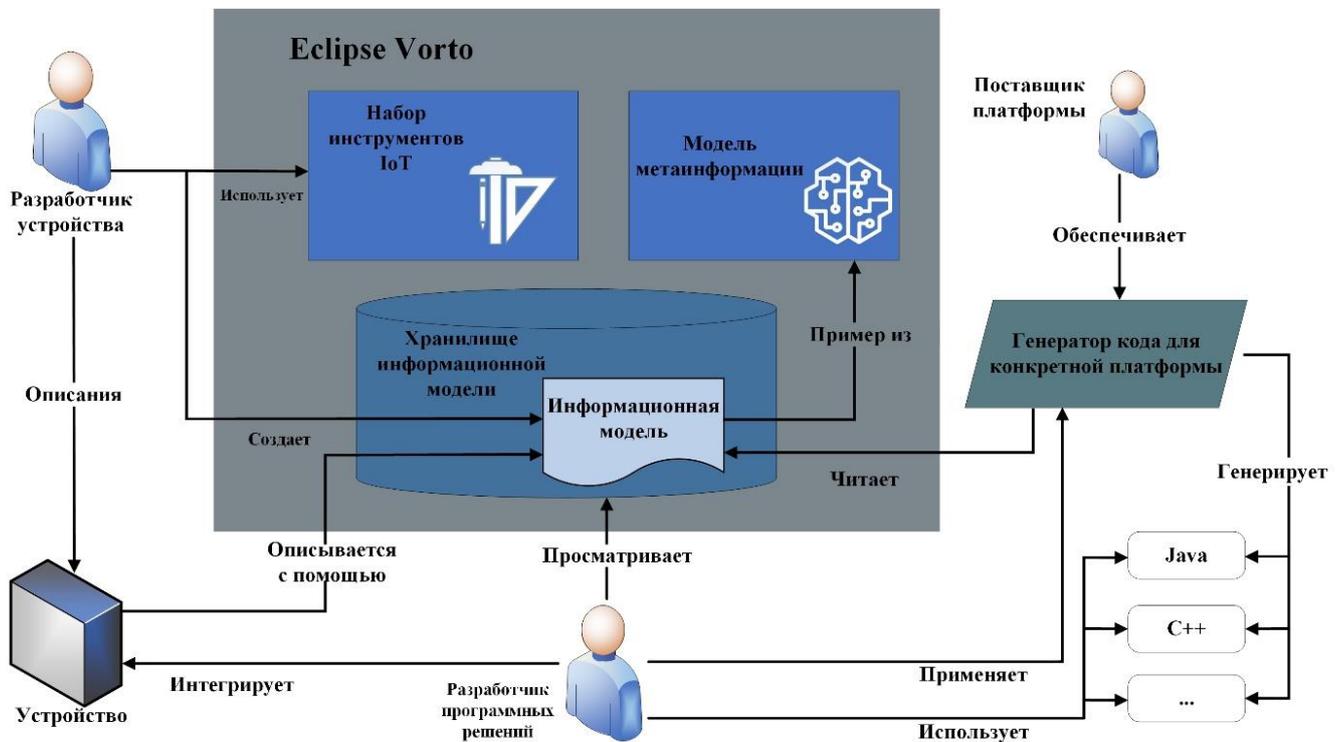


Рис. 3. Взаимодействие компонентов Eclipse Vorto в сочетании с соответствующими заинтересованными сторонами

Г. Платформа IoT и оператор приложения

Клиенты в среде IoT часто хотят использовать самые разные устройства и, следовательно, самые разные протоколы связи в сочетании. Поэтому различные устройства и их особенности также должны быть представлены в рамках одной платформы. При интеграции устройств это приводит к значительным усилиям по разработке. С генераторами кода Vorto эти усилия можно значительно сократить. Требуемый исходный код может быть сгенерирован либо полностью, либо хотя бы частично из описаний устройств производителя. Этот аспект позволяет поставщикам платформ реализовывать соответствующие генераторы кода. Хотя в краткосрочной перспективе здесь требуется больше усилий по разработке, в долгосрочной перспективе можно интегрировать все устройства, описания которых доступны в репозитории Vorto. Можно полагать, особенно с учётом быстро растущего разнообразия IoT, что такой подход окупится через короткое время.

Н. Организации по стандартизации

Модели данных играют важную роль в стандартизации. До сих пор для определения таких моделей часто использовались программы обработки текстов. Однако такой метод работы приводит к ошибкам при последующем изменении моделей. Это приводит к затратам на исправления, а каждая дополнительная разработка спецификации приводит к затратам на разработку. Eclipse Vorto предлагает способ значительно сократить такие усилия. Если модель указана в Eclipse Vorto, документ обзора может быть сгенерирован и вынесен на обсуждение. Разработчики могут обмениваться информацией онлайн через репозиторий Vorto и обсуждать ее непосредственно на модели. Результаты передаются в модель Vorto, после чего может быть сгенерирован новый обзорный документ. Этот процесс можно повторять сколько угодно раз, пока модель данных не будет принята.

IV. ECLIPSE VORTO ПРИМЕНИТЕЛЬНО К ЖЕЛЕЗНОДОРОЖНОМУ ТРАНСПОРТУ

Отслеживание передвижения вагонов и локомотивов, контроль технического состояния состава, прогнозирование предотказного состояния. Отказ от бумажного учета грузоперевозок и перевод всех документов в электронный формат. Обеспечение безопасности рабочего персонала, снижение влияния человеческого фактора в аварийных ситуациях. Вот

лишь часть возможностей, которые IoT устройства открывают для железной дороги.

Интернет вещей на железнодорожном транспорте – это в первую очередь датчики внутри и снаружи вагонов. Сейчас таких устройств используется всё больше, это направление активно развивается, но ему не хватает стандартов. Все датчики и системы считывания разнородные, построить единую систему сложно [6].

Чтобы разные устройства могли без проблем взаимодействовать друг с другом, предлагается использование проекта Eclipse Vorto. Данное программное решение позволяет не зависеть от широкого спектра протоколов связи, аппаратных сред и языков программирования различных производителей IoT устройств. Простота использования и интуитивно понятный интерфейс, вкуче с возможностью адресации и управления всеми устройствами через одну платформу, возможность гибкого и простого расширения сети IoT это то, что выгодно характеризует Eclipse Vorto.

V. ЗАКЛЮЧЕНИЕ И ДАЛЬНЕЙШИЕ ПЕРСПЕКТИВЫ ИССЛЕДОВАНИЯ

Благодаря гибкому подходу Eclipse Vorto облегчает и ускоряет распространение приложений IoT и решений от разных поставщиков в будущем. Платформа Eclipse Vorto сокращает усилия по разработке – благодаря высокой степени совместимости и гармонизации интерфейсов в области абстракции устройств, а также поддержке совместной работы между разработчиками IoT систем.

СПИСОК ЛИТЕРАТУРЫ

- [1] IoT Analytics. [Электронный ресурс]. – Режим доступа. – URL: <https://iot-analytics.com/> – Дата доступа: 15.02.2023
- [2] IoT Alliances and Consortiums. [Электронный ресурс]. – Режим доступа. – URL: <https://devopedia.org/iot-alliances-and-consortiums> – Дата доступа: 14.02.2023
- [3] Eclipse Vorto. [Электронный ресурс]. – Режим доступа. – URL: <https://www.eclipse.org/vorto/> – Дата доступа: 18.02.2023
- [4] Bosch Connected World Blog. [Электронный ресурс]. – Режим доступа. – URL: <https://blog.bosch-si.com/developer/eclipse-vorto-the-next-step-in-iot-device-integration/> – Дата доступа: 20.02.2023
- [5] Коршунов Н.С., Верба М.В. Анализ проблем безопасности интернета вещей // Международный журнал гуманитарных и естественных наук. 2019. № 2-1. С. 92-95.
- [6] Лаверман Д., Марко В. Интеграция данных о транспортных средствах в системы умного дома с помощью Eclipse Vorto // IEEE № 84 Конференция по транспортным технологиям (VTC-Fall), 2016, С. 20-26.